

REMARKS/ARGUMENTS

The claims have been amended as set forth above. Applicants believe the claims are in condition for allowance. Applicants respectfully request reconsideration of the claims.

I. Rejection Under 35 U.S.C. § 101

Claims 1 and 3 are rejected under 35 U.S.C. § 101 as including non-statutory subject matter. Independent claim 1 has been amended to recite a computer system comprising a processor, and a memory having computer-executable instructions. Accordingly, applicants assert that the 35 U.S.C. § 101 rejection has been overcome.

II. Rejection Under 35 U.S.C. § 103(a)

Claims 1, 3-8, 11-13 and 19-20 are rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 6,760,903 issued to Morshed (hereinafter “Morshed”) in view of U.S. Patent No. 4,866,599 issued to Morganti et al. (hereinafter “Morganti”). Applicants respectfully disagree with the rejection. Independent claim 1 has been amended to include the following combination of features that is not taught or otherwise suggested by the cited references:

a processor; and

a memory having computer-executable instructions stored thereon, wherein the computer executable instructions include:

a plurality of procedures, wherein each procedure comprises a sequence of binary instructions;

a runtime for generating unwind data, wherein the unwind data includes a first plurality of blocks of metadata having a first order of blocks, wherein each block of metadata is associated with a corresponding procedure in the plurality of procedures, wherein each block of metadata in the plurality of blocks of metadata includes at least one unwind table and at least one unwind information block; and

an unwind rewriter programmed to obtain the unwind data and reorder the first plurality of blocks of metadata to generate a second plurality of

blocks of metadata having a second order, wherein the unwind rewriter reorders the first plurality of blocks in accordance with a second unwind table and a second unwind information block, wherein reordering the first plurality of blocks in accordance with a second unwind table and a second unwind information block further comprises determining when basic blocks identified in a single unwind table associated with the first order of blocks are associated with more than one unwind table associated with a current order of basic blocks, creating a new region header describing a region of zero length when the basic blocks identified in the single unwind table associated with the first order blocks are associated with more than one unwind table associated with the current order of basic blocks, **and copying non-when action descriptor records from the unwind information block into the second unwind information block when the basic blocks identified in the single unwind table associated with the first order blocks are associated with more than one unwind table associated with the current order of basic blocks,** wherein the first plurality of blocks are reordered in response to a modification of the sequence of binary instructions within a procedure, such that the second plurality of blocks of metadata accurately represents the same runtime semantics as that of the unmodified sequence of binary instructions.

The cited references do not teach or otherwise suggest the above combination of features. Morshed pertains to gathering information associated with the distributed application. (Morshed at Abstract). The intermediate representation described in Morshed pertains to a typical intermediate representation during compilation of code. (Morshed at Col. 7, lines 49-57). The intermediate representation data is instrumented to provide object code. Morganti teaches a trap handling procedure in association with a ring architecture that determines a level privilege for execution of procedures related to each of the rings. A test procedure is implemented to cause the comparison of a first and second procedure to determine when a predetermined relationship exists between the first and second procedure. A stack is provided in association with each level of privilege for storing information related to procedures being executed at equivalent level of privilege. When a predetermined relationship exists between the first and second procedure attributes, control information in the second stack is entered. When the second procedure requires, data information is entered in the second stack that is under the processing system software control. Address information is entered to identify a location of control information in the first stack associated with the first procedure. The second procedure is then executed using

control information from the second stack. After the compilation of the second procedure, execution is returned to the first procedure by means of a procedure activated in response to the second procedure completion and executed under data processing system hardware control. Applicants can find no teaching or suggestion in either of the references of “copying non-when action descriptor records from the unwind information block into the second unwind information block when the basic blocks identified in the single unwind table associated with the first order blocks are associated with more than one unwind table associated with the current order of basic blocks.” Accordingly, applicants assert that independent claim 1 is in condition for allowance.

Independent claim 4 includes the following combination of features that is not taught or otherwise suggested by the cited references:

obtaining original unwind data that describes the original order of the basic blocks, wherein the original unwind data is associated with an unwind table and unwind descriptor records, **wherein the original order of basic blocks is noncontiguous and the original unwind data describes the noncontiguous order;**

regenerating new unwind data from the original unwind data, wherein regenerating new unwind data includes generating new unwind tables and new unwind descriptor records, **wherein each new unwind table pertains to a different grouping of contiguous basic blocks within the noncontiguous original order of basic blocks,** wherein the new unwind data includes a reordering of the original order of basic blocks, wherein regenerating the new unwind descriptor records further comprises determining when basic blocks identified in a single unwind table associated with the original order of basic blocks are associated with more than one unwind table associated with the current order of basic blocks, and creating a new region header describing a region of zero length when the basic blocks identified in the single unwind table associated with the original order of basic blocks are associated with more than one unwind table associated with the current order of basic blocks, wherein the reordering represents the same runtime semantics as that of the unmodified sequence of binary instructions, **and copying non-when action descriptor records from the original unwind descriptor records into the new unwind descriptor records;** and writing the new unwind data to the modified binary procedure.

The cited references do not teach or otherwise suggest the above combination of features. Morshed pertains to gathering information associated with the distributed application. (Morshed at Abstract). The intermediate representation described in Morshed pertains to a typical intermediate representation during compilation of code. (Morshed at Col. 7, lines 49-57). The intermediate representation data is instrumented to provide object code. Morganti teaches a trap handling procedure in association with a ring architecture that determines a level privilege for execution of procedures related to each of the rings. Neither reference teaches or otherwise suggests the combination of “wherein the original order of basic blocks is noncontiguous and the original unwind data describes the noncontiguous order,” in combination with “wherein each new unwind table pertains to a different grouping of contiguous basic blocks within the noncontiguous original order of basic blocks.” Furthermore, applicants can find no teaching or suggestion of “copying non-when action descriptor records from the original unwind descriptor records into the new unwind descriptor records.” Accordingly, applicants assert that independent claim 4 is in condition for allowance.

Independent claim 11 includes the following combination of features that is not taught or otherwise suggested by the cited references:

receiving unwind data comprising an unwind table and a plurality of unwind descriptor records wherein the unwind data is associated with a procedure having binary instructions;

modifying the procedure to perturb the binary instructions of the procedure;

parsing the unwind data to identify a start basic block and an end basic block for a region associated with the procedure;

rewriting the unwind data, wherein the rewriting of unwind data includes a reordering of unwind data in accordance with a second unwind table and a second plurality of unwind descriptor records such that the rewritten unwind data accurately represents the runtime semantics of the binary instructions before the binary instructions were perturbed, wherein generating the second plurality of unwind descriptor records further comprises determining when basic blocks identified in a single unwind table associated with the unmodified procedure are

associated with more than one unwind table associated with the binary modified procedure; and

when basic blocks identified in the unwind table are associated with more than one unwind table, creating a new region header describing a region of zero length and copying non-when action descriptor records from the unwind descriptor records into the second unwind descriptor records.

The cited references do not teach or otherwise suggest the above combination of features. Morshed pertains to gathering information associated with the distributed application. (Morshed at Abstract). The intermediate representation described in Morshed pertains to a typical intermediate representation during compilation of code. (Morshed at Col. 7, lines 49-57). The intermediate representation data is instrumented to provide object code. Morganti teaches a trap handling procedure in association with a ring architecture that determines a level privilege for execution of procedures related to each of the rings. Neither reference teaches or otherwise suggests “when basic blocks identified in the unwind table are associated with more than one unwind table, creating a new region header describing a region of zero length and copying non-when action descriptor records from the unwind descriptor records into the second unwind descriptor records.” Accordingly, applicants assert that independent claim 11 is in condition for allowance.

Independent claim 19 includes the following combination of features that is not taught or otherwise suggested by the cited references:

receiving unwind data comprising an unwind table and a plurality of unwind descriptor records wherein the unwind data is associated with a procedure having binary instructions;

modifying the procedure to perturb the binary instructions of the procedure;
parsing the unwind data to identify a start basic block and an end basic block for a region associated with the procedure;

rewriting the unwind data, wherein the rewritten unwind data includes a reordering of the unwind data according to a second unwind table and a second plurality of unwind descriptor records such that the rewritten unwind data accurately represents the runtime semantics of the binary instructions before the

binary instructions were perturbed, wherein reordering of the unwind data according to a second unwind table and a second plurality of unwind descriptor further comprises determining when basic blocks identified in the unwind table are associated with more than one unwind table associated with the binary modified procedure; and

when basic blocks identified in the unwind table are associated with more than one unwind table, creating a new region header describing a region of zero length and copying non-when action descriptor records from the unwind descriptor records into the second unwind descriptor records.

The cited references do not teach or otherwise suggest the above combination of features. Morshed pertains to gathering information associated with the distributed application. (Morshed at Abstract). The intermediate representation described in Morshed pertains to a typical intermediate representation during compilation of code. (Morshed at Col. 7, lines 49-57). The intermediate representation data is instrumented to provide object code. Morganti teaches a trap handling procedure in association with a ring architecture that determines a level privilege for execution of procedures related to each of the rings. Applicants can find no teaching or suggestion in either of the references of “when basic blocks identified in the unwind table are associated with more than one unwind table, creating a new region header describing a region of zero length and copying non-when action descriptor records from the unwind descriptor records into the second unwind descriptor records.” Accordingly, applicants assert that independent claim 19 is in condition for allowance.

Independent claim 20 includes the following combination of features that is not taught or otherwise suggested by the cited references:

obtaining original unwind data that describes the original order of the basic blocks, wherein the original unwind data is associated with an unwind table and unwind descriptor records, **wherein the original order of basic blocks is noncontiguous and the original unwind data describes the noncontiguous order;**

rewriting the original unwind data, wherein the rewritten unwind data includes a reordering of the original order of basic blocks, wherein rewriting the original unwind data includes:

parsing the original unwind data to identify a start block and an end block for region headers associated with the procedures in the modified binary procedures, wherein the identified start block and the identified end block are recorded in a procedure side table,

recording when action description records in the procedure side table,

generating new unwind tables based in the procedure side table, wherein each new unwind table pertains to a different grouping of contiguous basic blocks within the noncontiguous original order of basic blocks,

generating new unwind descriptors based on the procedure side table and the new unwind table,

reordering the original unwind data according to the new unwind table and the new unwind descriptors, wherein reordering includes determining when basic blocks identified in the new unwind table are associated with more than one unwind table,

when basic blocks identified in the new unwind table are associated with more than one unwind table, creating a new region header describing a region of zero length and copying non-when action descriptor records from the original unwind descriptor records into the new unwind descriptor records.

The cited references do not teach or otherwise suggest the above combination of features. Morshed pertains to gathering information associated with the distributed application. (Morshed at Abstract). The intermediate representation described in Morshed pertains to a typical intermediate representation during compilation of code. (Morshed at Col. 7, lines 49-57). The intermediate representation data is instrumented to provide object code. Morganti teaches a trap handling procedure in association with a ring architecture that determines a level privilege for execution of procedures related to each of the rings. Applicants can find no teaching or suggestion of “wherein the original order of basic blocks is noncontiguous and the original unwind data describes the noncontiguous order,” in combination with “generating new unwind tables based in the procedure side table, wherein each new unwind table pertains to a different grouping of contiguous basic blocks within the noncontiguous original order of basic blocks.” Applicants also can find no teaching or suggestion of “recording when action description records

in the procedure side table.” Also, applicants can find no teaching or suggestion within the references of “when basic blocks identified in the new unwind table are associated with more than one unwind table, creating a new region header describing a region of zero length and copying non-when action descriptor records from the original unwind descriptor records into the new unwind descriptor records.” Accordingly, applicants assert that independent claim 20 is in condition for allowance.

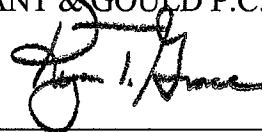
With regard to the dependent claims, the dependent claims include features that are not taught or otherwise suggested by the cited references. Furthermore, the dependent claims ultimately depend from the independent claims above. Accordingly, those claims should be found allowable for at least those same reasons.

III. Request for Reconsideration

In view of the foregoing amendments and remarks, all pending claims are believed to be allowable and the application is in condition for allowance. Therefore, a Notice of Allowance is respectfully requested. Should the Examiner have any further issues regarding this application, the Examiner is requested to contact the undersigned attorney for the applicant at the telephone number provided below.

Respectfully submitted,

MERCHANT & GOULD P.C.



RYAN T. GRACE

Registration No. 52,956

Direct Dial: 402.344.3000

MERCHANT & GOULD P.C.
P. O. Box 2903
Minneapolis, Minnesota 55402-0903
206.342.6200

